

PERANCANGAN SISTEM INFORMASI PENJUALAN, PEMBELIAN, DAN INVENTORI PADA CV BAMBI

Sugiarto Hartono

Computerized Accounting Department, School of Information Systems, Binus University
Jl. K.H. Syahdan No. 9, Palmerah, Jakarta Barat 11480
shartono@binus.edu

ABSTRACT

CV Bambi considers to use IS and IT to get competitive advantage. Thus, CV Bambi needs a system that utilizes IT and records all business transactions, especially in sales, purchasing, and inventory processes so that data redundancy will be minimal. This research creates information system design of sales, purchasing, and inventory processes as well as reports that can be used by the management. Methodology used in this paper is unified process with object-oriented approach. The result is an information system design that can solve problems in data searching since all the data have been stored in a database. When sales or purchasing transaction occurs, the stock in inventory system will be automatically updated. This system will make easy the manager to read reports by using reporting feature.

Keywords: *system design, sales system, purchasing system, inventory system*

ABSTRAK

CV Bambi sebagai perusahaan yang bergerak dalam bidang penjualan makanan ringan mulai menyadari bahwa penerapan SI dan TI sangat penting dalam pengembangan bisnisnya. Oleh karena itu, CV Bambi membutuhkan sebuah sistem yang memanfaatkan teknologi informasi yang mencatat seluruh transaksi bisnis yang berkaitan dengan sistem informasi penjualan, pembelian, dan inventori dalam sebuah database. Tujuannya agar data mudah dicari oleh unit yang berbeda dan redundansi data akan terminimalisasi. Penelitian ini dilakukan untuk membuat sebuah desain sistem informasi dari proses bisnis penjualan, pembelian, dan inventori yang pada akhirnya akan menghasilkan laporan yang dapat digunakan oleh pihak manajemen. Perancangan menggunakan metode unified process dengan pendekatan object oriented. Hasil dari penelitian ini adalah sebuah rancangan sistem informasi yang dapat mengatasi masalah dalam pencarian data pada proses penjualan, pembelian dan inventori karena semua data telah tersimpan dalam sebuah database. Dengan adanya sistem ini, seluruh data akan tersimpan dalam database dan terbarukan secara real time. Setiap terjadi transaksi penjualan, stok di sistem inventori akan langsung berkurang. Demikian halnya setiap terjadi pembelian barang, stok di inventori akan otomatis bertambah. Selain itu, sistem ini juga memudahkan manajer untuk mendapatkan laporan penjualan, pembelian, dan inventori melalui fitur laporan.

Kata kunci: *perancangan sistem, sistem penjualan, sistem pembelian, sistem inventori*

PENDAHULUAN

Perkembangan dunia bisnis terjadi sangat cepat. Seiring waktu, penggunaan Sistem Informasi (SI) dan Teknologi Informasi (TI) dalam pengembangan bisnis menjadi salah satu faktor utama untuk meningkatkan kualitas kinerja perusahaan dalam menjalankan rutinitas proses bisnis. Dengan semakin banyaknya perusahaan yang berlomba-lomba untuk mengimplementasikan TI dalam perusahaan mereka, kebutuhan SI dan TI ikut berkembang pesat.

Sistem informasi harus dapat menyediakan informasi untuk orang yang tepat dalam waktu yang tepat serta dalam format dan jumlah yang tepat dan sesuai dengan kebutuhan penerima informasi (Rainer & Cegielski, 2011:10). Sistem informasi akan memberikan manfaat bagi *user*, sistem, organisasi, dan level strategis di dalam perusahaan (Peterson & Kim, 2000:29). Berdasarkan fungsinya, sistem informasi dapat diklasifikasikan dalam beberapa jenis, yaitu: *transaction processing systems*, *management information system*, *executive information systems*, *decision support systems*, *communication support systems*, dan *office support systems* (Whitten & Bentley, 2007:6).

Sebuah perusahaan harus memiliki *transaction processing system* yang baik sebelum dapat mengimplementasikan jenis sistem informasi lainnya. *Transaction Processing Systems* adalah sistem informasi yang menangkap dan merekam informasi transaksi yang mempengaruhi organisasi, misalnya transaksi order, pembayaran, dan sebagainya (Whitten & Bentley, 2007:6). *Scope transaction processing system* yang dibahas dalam penulisan ini adalah sistem penjualan, pembelian, dan inventori. Ketiga sistem tersebut merupakan sistem utama dari setiap perusahaan yang bergerak dalam bidang *trading* untuk dapat mengoperasikan proses bisnisnya.

Sistem penjualan, pembelian, dan inventori perusahaan yang didukung dengan penggunaan SI dan TI secara maksimal dapat meningkatkan keunggulan secara kualitatif dan kuantitatif. Dilihat dari segi kualitatif, penggunaan SI dan TI dapat meningkatkan efektifitas proses bisnis perusahaan dan meningkatkan kepuasan pelanggan. Dan dilihat dari segi kuantitatif, penggunaan SI dan TI dapat meningkatkan keuntungan ekonomi bagi perusahaan tersebut. Desain sistem informasi yang efektif harus dapat digunakan dan memberikan pengaruh positif bagi kepentingan organisasi (Markus, Majchrzak, & Gasser, 2003:179).

Objek penelitian yang digunakan adalah CV Bambi. Perusahaan ini bergerak dalam bidang penjualan *Food & Beverages* dengan spesialisasi penjualan makanan ringan (*snack*). Perusahaan ini menganggap bahwa penerapan SI dan TI sangat penting dalam pengembangan bisnis. Namun perusahaan belum menerapkan teknologi informasi ke dalam setiap detail proses bisnisnya. Oleh karena itu, CV Bambi membutuhkan sebuah sistem yang memanfaatkan teknologi informasi yang mencatat seluruh transaksi bisnis yang berkaitan dengan sistem informasi penjualan, pembelian, dan inventori dalam sebuah *database* sehingga data akan mudah dicari oleh unit yang berbeda. Selain itu, redundansi data akan terminimalisasi. Tujuan dari penelitian ini adalah membuat sebuah desain sistem informasi dari proses bisnis penjualan, pembelian, dan inventori pada CV Bambi, di mana sistem tersebut akan menghasilkan laporan yang dapat digunakan oleh pihak manajemen untuk keperluan strategis.

METODE

Metode yang digunakan dalam penelitian ini adalah: (1) observasi, yaitu pengamatan dan penelitian pada CV Bambi untuk mendapatkan data dan informasi yang dibutuhkan untuk merancang

sistem informasi; (2) wawancara, yaitu sesi tanya jawab kepada pihak CV Bambi yang terkait untuk mengetahui proses bisnis yang berjalan serta untuk mengidentifikasi kebutuhan *user*; (3) analisis sistem pada data-data yang didapat dengan menggunakan metode *unified process* berorientasi objek.

Unified Process merupakan salah satu metodologi untuk melakukan *problem solving* dan membutuhkan adanya peran serta partisipasi seluruh *stakeholder* (Forster & Brocco, 2008). *Stakeholder* dalam hal ini adalah pihak pengembang sistem dan pihak perusahaan CV Bambi. Keseluruhan *stakeholder* tersebut melakukan *brainstorming* untuk menganalisis sistem yang akan dikembangkan. Analisis sistem dalam penelitian ini menggunakan pendekatan *Object-Oriented*, di mana untuk penggambaran model analisis menggunakan notasi grafis UML.

Selanjutnya adalah perancangan sistem sesuai dengan analisis yang telah dilakukan. Perancangan sistem ini menggunakan perancangan metode berorientasi objek. Tahapan desain sistem yang dilakukan yaitu: (1) desain *support services architecture* dan *deployment environment*; (2) desain *software architecture*; (3) desain *usecase realization*; (4) desain *user interface*; (5) desain *system security and control* (Satzinger, Jackson, & Burd, 2012:162).

HASIL DAN PEMBAHASAN

Saat ini sistem penjualan pada CV Bambi masih menggunakan sistem manual. Di bawah ini adalah proses bisnis yang diusulkan setelah menggunakan sistem komputerisasi:

Pemilik melakukan perekrutan agen yang sebelumnya melakukan survei terlebih dahulu. Setelah menemukan agen, pemilik akan mencatat data agen serta agen memberikan surat jaminan. Kemudian, pemilik akan memasukkan data agen dan surat jaminan ke dalam sistem.

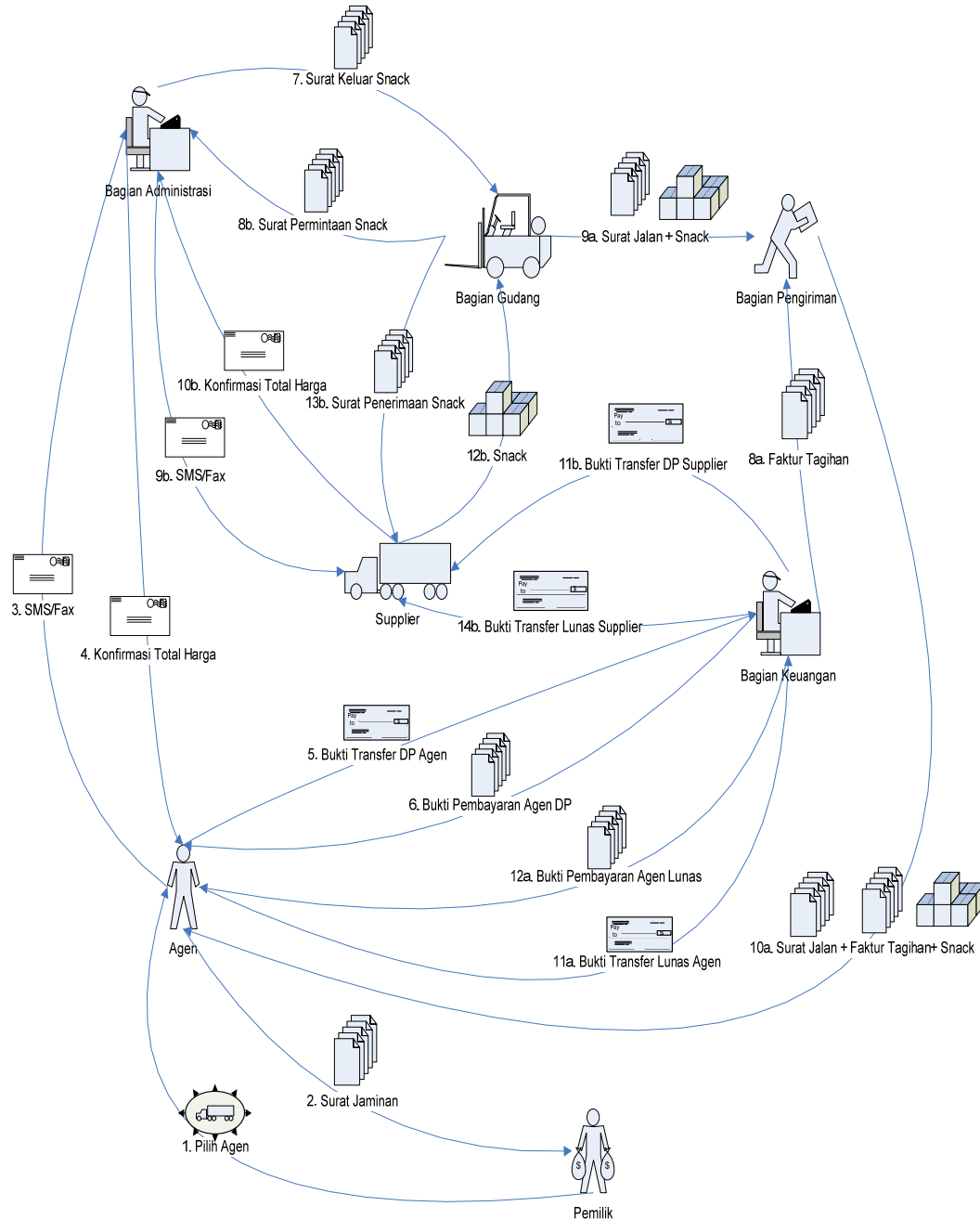
Bagian administrasi menerima pemesanan dari agen melalui SMS yang berisi nama dan jumlah *snack*. Pemesanan tersebut akan dimasukkan oleh bagian administrasi ke dalam nota pemesanan dengan catatan agen harus membayar *down payment* minimal 15% dari total harga. Setelah agen membayar *down payment* maka bukti pembayaran agen (DP) akan dibuat oleh bagian keuangan. Setelah itu bagian administrasi akan membuat surat keluar *snack* yang berisi nama dan jumlah *snack* yang ditujukan ke bagian gudang.

Setelah menerima surat keluar *snack* dari bagian administrasi, bagian gudang akan mengecek ketersediaan jumlah *snack* yang ada. Apabila ketersediaan *snack* digudang mencukupi, bagian gudang akan membuat surat jalan dan bagian keuangan akan membuat faktur tagihan. Apabila tidak mencukupi, bagian gudang akan membuat surat permintaan *snack* yang berisi nama *snack* yang ketersediaannya melewati batas minimum ke bagian administrasi. Bagian keuangan bertugas membuat faktur tagihan yang akan diberikan ke agen. Faktur tagihan tersebut berisi total tagihan yang belum dibayarkan oleh agen. Surat jalan yang dibuat oleh bagian gudang akan diserahkan ke bagian pengiriman beserta faktur tagihan yang telah dibuat oleh bagian keuangan.

Bagian keuangan menerima bukti sisa pembayaran melalui faks yang dikirimkan oleh agen. Apabila agen telah melakukan pelunasan pembayaran, bagian keuangan akan membuat bukti pembayaran agen (lunas) dan akan dikirimkan kembali ke agen melalui faks. Setelah menerima surat permintaan *snack* dari bagian gudang, Bagian administrasi akan melakukan pemesanan *snack* ke *supplier* dengan cara mengirimkan SMS kepada *supplier*. Pemesanan tersebut akan dicatat dan disimpan ke *purchase order*. Setelah bagian administrasi melakukan pemesanan, *supplier* akan mengirimkan *snack* yang dipesan.

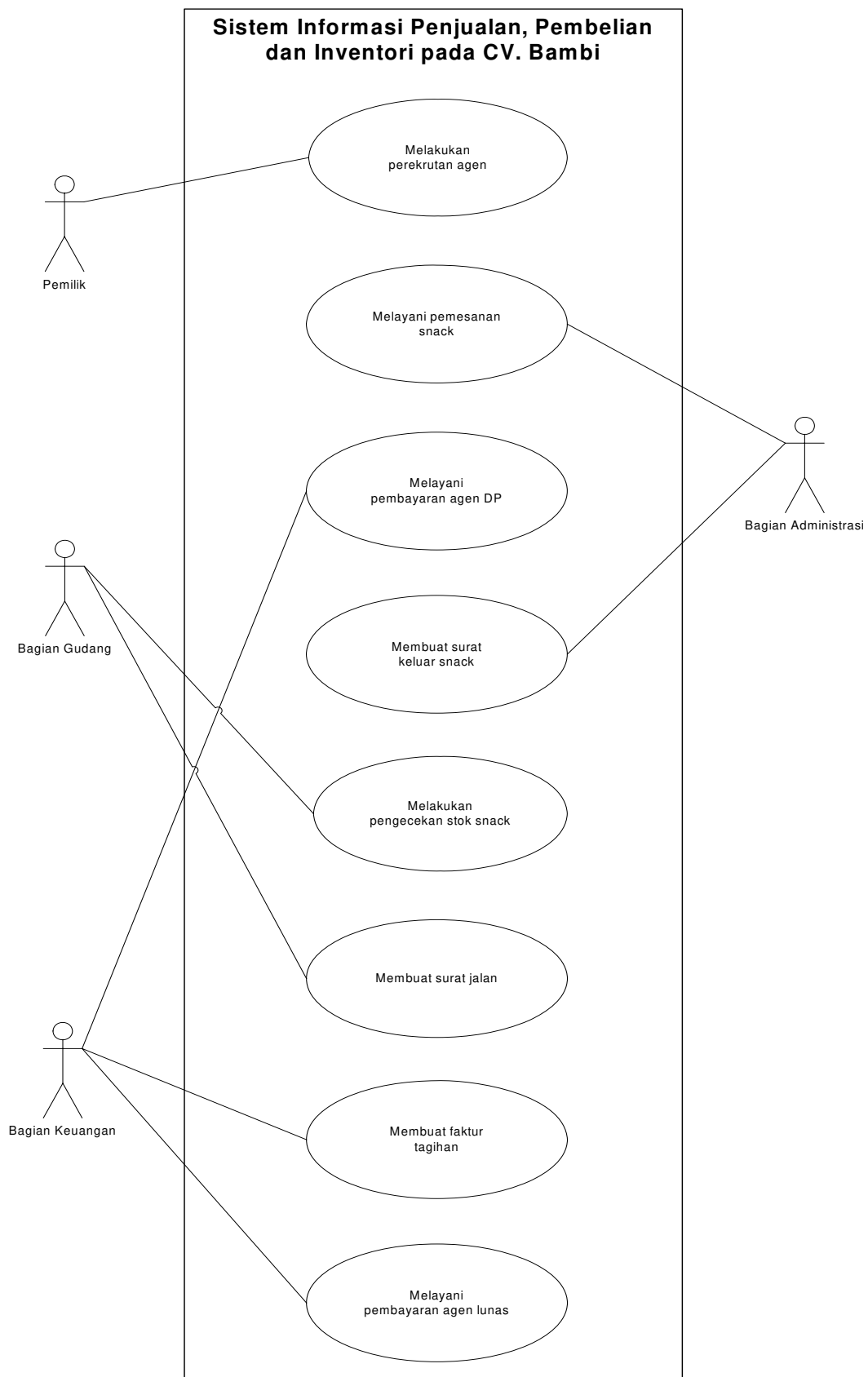
Barang yang telah dikirim oleh *supplier* akan diterima oleh bagian gudang dan akan dibuat surat penerimaan barang. Setelah barang diterima, bagian keuangan akan melakukan pembayaran *down payment* sebesar 15% dari total harga kepada *supplier* dan akan dimasukkan ke dalam bukti pembayaran *supplier* (DP). Dalam jangka waktu tertentu bagian keuangan akan melakukan pelunasan pembayaran kepada *supplier* dan disimpan ke dalam bukti pembayaran *supplier* (lunas).

Seluruh proses di atas dirangkum pada rich picture di bawah ini (Gambar 1).



Gambar 1 Proses bisnis penjualan, pembelian, dan inventori yang diusulkan.

Functional requirement digambarkan menggunakan *usecase diagram* seperti berikut ini (Gambar 2):

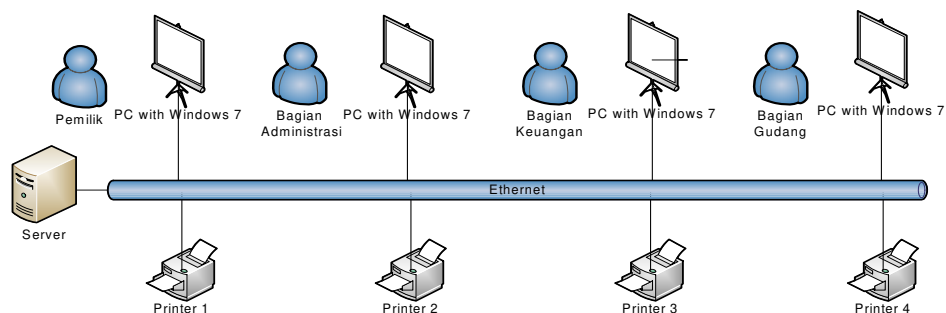


Gambar 2 Usecase diagram



Gambar 2 Usecase Diagram (Lanjutan)

CV Bambi menggunakan *deployment architecture* (Gambar 3) dengan tipe *multitier architecture* yang memungkinkan kegiatan *computing* terjadi di beberapa komputer. Hal tersebut dapat dilihat dari sistem yang berjalan di CV Bambi cukup banyak, meliputi proses pembayaran ke agen dan *supplier*, *update* stok *snack*. CV Bambi menjalankan semua sistem menggunakan tipe *multicomputer architecture*, dengan pembagian beban melalui spesialisasi fungsi. Oleh karena itu, tiap komputer yang digunakan memiliki spesifikasi yang berbeda yang disesuaikan berdasarkan fungsi dan peran nya masing-masing. Walaupun setiap bagian memiliki fungsi dan peran yang berbeda, semua bagian tetap saling terhubung satu sama lain. Makad adri itu ehingga suatu pekerjaan dapat direspon dengan cepat oleh bagian lain yang akan melanjutkan pekerjaan tersebut. Hal ini dapat terjadi karena setiap bagian sudah tersambung oleh *server*. CV Bambi menempatkan semua sumber penghitungan pada tempat yang terpusat, karena CV Bambi tidak memilik cabang di tempat lainnya, sehingga tidak memerlukan jaringan komputer untuk menghubungkan.



Gambar 3 Deployment environment

CV Bambi menggunakan *two tier* pada *software architecture*, karena setiap bagian menggunakan aplikasi yang sama pada setiap komputernya, dan dapat mengakses ke *database* yang sama karena memiliki *server* yang sama dan saling terhubung satu sama lain. Di bawah ini adalah class diagram yang dirancang (Gambar 4). Rancangan *User Interface* yang diusulkan sebagai berikut:

Untuk membuat nota pemesanan (Gambar 5), Bagian administrasi harus memilih *button Create New* terlebih dahulu. Lalu dimulai dari pemilihan agen yang memesan. Ketika *textbox* ID agen diklik, akan muncul *form* yang menampilkan data agen. Berikutnya, bagian administrasi memilih *snack* yang dipesan oleh agen, dan *list snack* yang dipesan oleh agen akan ditampilkan di *datagrid* detail nota pemesanan. Jika terdapat kesalahan dalam penginputan data *snack*, bagian administrasi bisa menggunakan *button reset* untuk menghapus data-data yg ada di *datagrid*, lalu bagian administrasi bisa memasukkan ulang *snack* yang ingin dipesan. Setelah selesai, bagian administrasi dapat memilih *button Save and Print* untuk menyimpan dan mencetak Nota Pemesanan yang dibuat. *Button Cancel* digunakan untuk menutup *form* nota pemesanan.

Untuk membuat surat keluar *snack* yang baru (Gambar 6), Bagian administrasi harus memilih *button Create New* terlebih dahulu. Dimulai dari pemilihan nota pemesanan. Ketika *textbox* KD Nota Pemesanan terisi, lalu nama agen akan terisi secara otomatis. Berikutnya, bagian administrasi bisa melihat *list snack* yang sudah dipesan agen di *datagrid*. Setelah selesai, bagian administrasi dapat memilih *button Save and Print*, maka surat keluar *snack* akan tersimpan dan muncul *Print Preview*. *Button cancel* digunakan untuk menutup *form* surat keluar *snack*.

Untuk membuat *purchase order* (Gambar 7), Bagian administrasi harus memilih *button Create New* terlebih dahulu yang baru. Berikutnya, bagian administrasi memilih *supplier* yang akan menyuplai *snack*, dan memilih kd surat permintaan *snack*. *List snack* yang dipesan pun akan ditampilkan di *datagrid* detail surat permintaan *snack*. Setelah selesai, bagian administrasi dapat memilih *button Save and Print*, untuk menyimpan dan mencetak *purchase order*. *Button cancel* digunakan untuk menutup *form purchase order*.

Untuk membuat laporan penjualan (Gambar 8), Bagian Administrasi pertama kali harus memilih periode laporan penjualan. Bagian administrasi bisa memilih laporan penjualan ingin ditampilkan berdasarkan nama agen atau nama *snack*. Setelah itu, bagian administrasi harus memilih *button OK* untuk menampilkan data transaksi penjualan selama periode yang telah ditentukan. Lalu, bagian administrasi bisa memilih untuk mencetak dengan memilih *button Print*.

Form Nota Pemesanan

KD Nota Pemesanan : Tanggal : 12/07/2011

ID Agen :
 Nama Agen :

Snack
 Search by Nama Snack

ID Snack	Nama Snack	Stok Snack	Harga Jual

ID Snack :
 Nama Snack :
 Harga Snack : Rp
 Jumlah Pemesanan :

Detil Nota Pemesanan

ID Snack	Nama Snack	Harga Snack	Jumlah Snack	Subtotal

Add
Remove

Total Harga Rp 0

Create New Save and Print Reset Cancel

Gambar 5 Rancangan *form* nota pemesanan

Form Surat Keluar Snack

KD Surat Keluar Snack : Tanggal : MM/dd/yyyy

KD Nota Pemesanan :
 Nama Agen :

Detil Nota Pemesanan

ID Snack	Nama Snack	Jumlah Pemesanan

Create New Save and Print Cancel

Gambar 6 Rancangan *Form* surat keluar *snack*

Gambar 7 Rancangan *Form Purchase Order*

Gambar 8 Rancangan *Form laporan penjualan*

Berikut ini adalah *Input integrity control* yang akan diimplementasikan pada CV Bambi.

Input Integrity Controls

Field Combination Controls

Pada sistem yang di usulkan pada CV.Bambi terdapat *field combination control* yaitu *integrity control* yang memverifikasi data di suatu *field* dengan berdasarkan pada data yang berada di *field* lain. Contohnya, apabila bagian administrasi ingin membuat surat keluar snack kepada bagian gudang maka bagian administrasi membutuhkan data yang ada di nota pemesanan. Jadi ketika bagian administrasi membuat surat keluar *snack* maka data nota pemesanan akan muncul di *form* surat keluar *snack*.

Value Limit Controls

Value limit control adalah *integrity control* yang mengidentifikasi bahwa nilai yang dimasukkan pada *field* masuk akal. Contohnya, apabila persediaan *snack* yang ada di gudang tersisa

500, bagian gudang harus membuat surat permintaan *snack*. Surat permintaan *snack* tidak bisa dibuat apabila jumlah *snack* yang berada digudang lebih dari 1000.

Completeness Controls

Completeness controls adalah *integrity control* yang memastikan bahwa semua *field* yang ada di suatu sudah terisi semua. Misalnya, saat pemilik ingin membuat data agen baru maka pemilik harus memasukkan nama, alamat, dan nomor telepon agen. Apabila salah satu *field* belum terisi maka ketika pemilik memilih tombol *save* maka akan keluar peringatan bahwa salah satu *field* belum terisi.

Data Validation Controls

Data validation control dilakukan dengan verifikasi *field* numerik yang berisi kode sudah diisi dengan benar. Contohnya, apabila bagian administrasi ingin *login*, Bagian administrasi harus memasukkan *username* dan *password* yang panjang nilainya sudah ditentukan dan sesuai dengan yang terdapat pada *database*.

Database Integrity Controls

Access Controls

Integrity control yang membatasi siapa saja yang dapat mengakses sistem. Misalnya yang hanya dapat melihat, menghapus dan mengubah jumlah ketersediaan *snack* yang ada digudang adalah bagian gudang.

Transaction Controls

Teknik *integrity control* yang apabila *database* di-*upgrade* akan dicatat sesuai misalnya dengan siapa yang melakukan *update*.

Update Controls

Update control adalah *integrity control* di mana apabila suatu *form* akan di *upgrade*, *field* yang dapat diubah hanya *field* selain dari *primary key*. Apabila ingin mengubah data agen, pemilik tidak dapat mengubah ID pemilik.

Output Integrity Controls

Destination Controls

Form satu dengan *form* yang lainnya saling terkait atau berhubungan dan ditujukan ke bagian tertentu. Misalnya bagian administrasi membuat surat permintaan barang ke bagian gudang dan *field* yang ada di di *form* surat permintaan barang sebagian berisi *field* yang ada di nota pemesanan.

Completeness, Accuracy dan Correctness Control

Setiap laporan yang dibuat oleh bagian administrasi dan bagian gudang di setiap bulannya akan tertera tanggal di hasil cetaknya dan akan diperjelas dengan ditambahkan *bold*.

Setiap bagian yang ada diperusahaan ingin melakukan transaksi harus melakukan *login* terlebih dahulu. Setiap bagian memiliki kode bagian masing-masing. Pembuatan kode khusus ditiap bagian dimaksudkan untuk meningkatkan keamanan data transaksi.

User CV Bambi di dalam aplikasi dibagi menjadi tiga kategori, yaitu *unauthorized user*, *registered user* dan *privileged use*. Bagian pengiriman, agen dan *supplier* termasuk *unauthorized user*. Ketiganya tidak diperbolehkan untuk mengakses ke dalam sistem. Agen hanya dapat melakukan pemesanan melalui SMS, begitupun dengan *supplier* yang tidak bisa mengakses ke dalam sistem. Bagian pengiriman hanya mendapat tugas untuk mengirimkan pesanan dan tidak diperbolehkan untuk mengakses sistem. *Registered user* adalah orang yang dapat mengakses ke dalam sistem. Pemilik, bagian administrasi dan bagian gudang termasuk dalam kategori *registered user*. Ketiganya dapat mengakses ke dalam sistem untuk melakukan transaksi di bagian masing-masing. *Privileged user* dapat disebut juga sebagai *programmer* atau orang yang mengerti seluruh sistem komputerisasi dan keamanan data yang berjalan. Yang termasuk dalam *privileged user* adalah pemilik karena pemilik akan diberikan pelatihan untuk dapat mengakses program dan *database* yang ada di perusahaan. Bagian administrasi juga dapat dimasukkan ke dalam kategori ini karena bagian administrasi yang akan menjalankan banyak proses bisnis yang terjadi di dalam perusahaan.

PENUTUP

Dari penelitian ini dapat disimpulkan bahwa CV Bambi masih menggunakan proses secara manual pada proses bisnis penjualan, pembelian dan inventori. Pencatatan data penjualan, pembelian dan inventori yang dilakukan di kertas memungkinkan data tercecer dan hilang. Untuk membuat laporan penjualan, pembelian dan inventori harus dilakukan rekapitulasi yang menghabiskan waktu banyak seperti harus mengumpulkan semua nota pemesanan dari awal sampai akhir bulan. Hasil dari perancangan sistem informasi ini adalah sebuah sistem informasi yang dapat mengatasi masalah dalam pencarian data baik pada proses penjualan, pembelian ataupun inventori karena semua data telah terintegrasi satu dengan yang lainnya dan terupdate secara *realtime*. Melalui fitur *reporting* pada sistem akan mempermudah pemilik perusahaan untuk melihat laporan penjualan, pembelian dan inventori setiap bulannya tanpa perlu melakukan rekapitulasi transaksi-transaksi yang terjadi.

DAFTAR PUSTAKA

- Forster, F., Brocco, M. (2008). Understanding Creativity-Technique Based Problem Solving Processes. *12th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, 5178. New York: Heidelberg.
- Markus, M. L., Majchrzak, A., & Gasser, L. (2003). A Design Theory for Systems that Support Emergent Knowledge Processes. *MIS Quarterly*, 26, 179 – 212.
- Mathiassen, L., Munk-Madsen, A., Nielsen, P.A., Stage, J. (2000). *Object-Oriented Analysis & Design*. Denmark: Marko Publishing.
- Peterson, D.K. & Kim, C.S. (2000). Information system objectives: effects of experience, position level, and education on developers. *Journal of Information Technology Management*, 11 (3 – 4).
- Rainer, R.K. & Cegielski, C.G. (2011). *Introduction to Information System*. New Jersey: John Wiley & Sons.
- Satzinger, J W., Jackson, R.B, & Burd, S.D. (2012). *System Analysis and Design in a Changing World*. Kentucky: Cengage Learning.
- Whitten, J.R. & Bentley, L.D. (2007). *Systems Analysis and Design Method*. New York: McGraw-Hill.